

TrustFV: Trustworthy Fact Verification on Knowledge Graph via Process Reward-Guided Step-Level Constrained Decoding

Yue Zhang College of Computer Science and Technology, National University of Defense Technology. zhangyue@nudt.edu.cn	Shicheng Zhou Laboratory for Advanced Computing and Intelligence Engineering. zhoushicheng@nudt.edu.cn	Yifu Gao * College of Computer Science and Technology, National University of Defense Technology. Gaoif@nudt.edu.cn
Zhiliang Tian * College of Computer Science and Technology, National University of Defense Technology. tianzhiliang@nudt.edu.cn	Wenqing Hou College of Computer Science and Technology, National University of Defense Technology. wenqinghou@nudt.edu.cn	Bin Zhou College of Computer Science and Technology, National University of Defense Technology. binzhou@nudt.edu.cn

Abstract

The rapid spread of misinformation has made fact verification a critical necessity. LLM-based methods have become the mainstream approach for fact verification, but they still suffer from hallucinations. To mitigate hallucinations, researchers incorporate KG on LLM reasoning with a graph-based constrained decoding. However, these methods with constrained decoding suffer from **error cascading** due to the inherent knowledge gaps of the LLM, which cause it to select wrong branches during beam search in the constrained token space. To tackle **error cascading**, we think that the decoding process should be dynamically guided by step-level plausibility assessment. In light of this, we propose TrustFV¹, a fact verification framework that step-wise beam search in token-level constrained decoding with risk-averse pruning for trustworthy verification. Specifically, TrustFV integrates (1) token-level constrained decoding on a Knowledge Graph Tries (KG-Tries) to ground generation in retrieved evidence from KG, (2) an Uncertainty-aware Process Reward Model (UPRM) trained via Monte Carlo roll-outs and ordinal regression to estimate both expected success and uncertainty per step, and (3) a step-wise beam search that actively prunes high-risk branches via the step-level signal of UPRM, producing faithful and plausible reasoning paths. Experiments show that TrustFV achieves state-of-the-art performance on FactKG and CWQ datasets.

1 Introduction

The rapid expansion of misinformation has made Fact Verification (FV) a crucial task. FV automat-

ically judges whether claims are supported or refuted by credible evidence (Bekoulis et al., 2023). Traditional FV works rely on textual evidence (Lee et al., 2021; Schuster et al., 2021), causing implicit reasoning that is hard to verify. In contrast, Knowledge Graphs (KGs) structure evidence as triples (Kim et al., 2023b), providing explicit topological constraints that lay the foundation for rigorous and interpretable reasoning.

Existing KG-based FV methods can be broadly divided into two categories: (1) Graph-based classification methods; (2) Large Language Models (LLMs) reasoning-based methods. Graph-based classification methods (Chen et al., 2025; Zhou et al., 2019) use Graph Neural Networks (GNNs) to encode evidence subgraphs into latent embeddings for binary classification. While effective in predicting labels, these methods are limited in their capacity to produce traceable reasoning paths. For detailed reasoning paths, researchers harness the strong reasoning and generation capabilities of LLMs for reasoning paths generation, called LLM-based reasoning methods (Cheng et al., 2024; Cai et al., 2025). They use LLMs to construct triple-format reasoning programs from claims and execute sequential verification over evidence subgraphs (Hao and Wu, 2025; Luo et al., 2024), thereby generating explicit reasoning traces. However, LLMs are prone to structural hallucinations that introduce triples inconsistent with KG, thereby producing reasoning paths that are not grounded in the KG.

To eliminate structural hallucination, pioneering works (Luo et al., 2025; Li et al., 2025) combine graph-based and LLM-based methods, which typ-

* Corresponding authors.

¹https://github.com/2twentytwo2/Trust_KG_FV

ically involve KG on LLM reasoning with graph-constrained decoding. By confining the decoding token space within the KG, these methods ensure that generated reasoning paths are consistent with the KG, successfully eliminating structural hallucinations (Luo et al., 2025). However, a major drawback of these works is **error cascading**. During beam search in the strict token space, these approaches rely solely on the LLM’s raw logits for branch selection. Due to inherent knowledge gaps, the LLM frequently assigns high probabilities to logically flawed branches, and current decoding methods cannot recognize and correct them.

To tackle **error cascading** in constrained decoding methods, we rethink the decoding process from a “look-before-you-leap” perspective. Our core insight is that the constrained decoding process should be dynamically guided by step-wise plausibility. By infusing progressive step-evaluation into the decoding process, we proactively steer the reasoning paths away from erroneous branches, ensuring that conclusions are derived exclusively from trustworthy paths within the KG.

To operationalize this insight, we propose TrustFV, a novel framework for trustworthy fact verification that utilizes process reward-guided step-level constrained decoding on KG. TrustFV dynamically evaluates the plausibility of each candidate reasoning step as it is generated on Knowledge Graph Tries (KG-Tries), actively pruning high-risk branches that are contradictory or lack sufficient confidence. Consequently, every retained step is not only grounded in the KG but also plausible, ensuring the trustworthiness of the fact verification process. Specifically, we first construct a KG-Trie on subgraphs. Furthermore, to identify high-risk steps during reasoning, we propose an Uncertainty-aware Process Reward Model (UPRM), which learns the success probability distribution via Monte Carlo rollouts and ordinal regression, for step-level assessment. Based on the assessment, we prune high-risk steps actively to tackle error cascading during step-wise beam search for a trustworthy reasoning process on KG.

Our key contributions are threefold: (1) We propose TrustFV, a fact verification framework over KG that integrates token-level constrained decoding with step-level risk-averse pruning. TrustFV evaluates each reasoning step dynamically, proactively pruning erroneous branches to suppress error cascading. (2) We design a UPRM to quantify both the success probability and the uncertainty per rea-

soning step, and couple it with a risk-averse beam search that prunes unreliable branches during decoding, ensuring grounded and plausible reasoning paths. (3) Experiments on two datasets demonstrate that TrustFV achieves state-of-the-art (SOTA) performance through trustworthy reasoning paths.

2 Related Works

2.1 Fact Verification based on KG

To overcome the redundancy and noise in plain-text evidence, researchers have introduced KG for FV (Kim et al., 2023b), leveraging its explicit relational constraints and high-fidelity evidence graphs.

Early methods employ GNNs (Zhou et al., 2019; Chen et al., 2025) to encode and aggregate evidence subgraphs for veracity classification. However, relying on high-dimensional vector aggregation, they cannot produce traceable reasoning paths. To enable explicit reasoning, recent work has adopted LLMs (Cai et al., 2025; Kim et al., 2023a; Cheng et al., 2024), decomposing verification into multiple triple-level steps for incremental matching and inference (Hao and Wu, 2025; Luo et al., 2024). However, lacking KG structural awareness, LLMs tend to generate non-existent triples, called reasoning hallucinations. To eliminate such hallucinations, prior works proposed constrained decoding on KG-Tries (Luo et al., 2025; Li et al., 2025), which strictly restricts generation to valid KG paths by masking out any tokens that fall outside the KG. But a limitation of the decoding methods is that non-useful branches in the KG-Trie mislead the model, and fail to remedy such error cascading.

To tackle the error cascading issue, we use step-level assessment, performing an immediate check as each reasoning step is generated and pruning high-risk branches to block errors at their source.

2.2 Process Reward Model

Process reward models (PRMs) (Zheng et al., 2025) quantitatively evaluate each step of a reasoning path to ensure that the final answer results from rigorous and logically sound derivation at every step. Early PRMs relied on expensive manual step-level annotation. To overcome this bottleneck, research shifted toward automated synthesis methods. Currently, automated approaches such as Monte Carlo tree search (MCTS)-based trajectory rollouts have become mainstream for generating large-scale PRM supervision data (Tian et al., 2024; Wang et al., 2024). By repeatedly sampling from inter-

mediate nodes and computing the empirical success rate, these methods convert hard labels into continuous probability values (Park et al., 2025) that faithfully reflect the likelihood of eventual correctness. This shift has enriched the fine-grained semantics of step rewards and driven the evolution of PRMs from binary classification toward multi-dimensional evaluation (Zhao et al., 2026; Yin et al., 2025). Motivated by this trend, our work adopts distributional learning to model the probability distribution of success rates, quantifying each step’s success probability and uncertainty via the distribution’s expectation and spread.

3 Methods

3.1 Overview

To overcome the **error cascading** caused by blind beam expansion in KG-based FV, we propose TrustFV, which uses step-wise pruning in graph-constrained reasoning for trustworthy reasoning paths. TrustFV (Fig. 1) has three components: **1. Token-level Constrained Decoding Mechanism** (§3.2) constructs a generated KG-Trie to support disconnected subgraphs retrieved from KG, and applies constrained decoding to restrict generated tokens within the KG-Trie for a faithful reasoning process on subgraphs. **2. Uncertainty-aware Process Reward Model** (§3.3) learns a discrete distribution over success rates via Monte Carlo roll-outs and ordinal regression, enabling detailed step-level estimation of both expected success and uncertainty. **3. Step-wise Beam Search for Trustworthy Reasoning** (§3.4) uses the step-level assessment of UPRM to actively prune high-risk branches during LLM beam searching on a KG-Trie, thus the LLM yields a grounded and plausible reasoning path.

3.2 Token-level Constrained Decoding Mechanism

To ensure that the LLM reasons strictly from KG, we design a constrained decoding mechanism based on retrieved subgraphs from KG. Unlike existing methods, where each branch in the KG-Trie stores only a consecutive sequence of head-to-tail triples (Li et al., 2025; Luo et al., 2025), we extend the branch structure to represent disconnected subgraphs 3.2.1. Based on the generated KG-Trie, the LLM generates a faithful reasoning process 3.2.2, which is grounded in the retrieved evidence, thereby eliminating structural hallucinations.

3.2.1 Generalized KG-Tries Construction on Subgraphs

In fact verification, evidence is often scattered across topologically disconnected subgraphs in KG. To address this, we model a generalized KG-Trie by extending the KG-Trie branch structure to support disconnected subgraphs.

Given a claim and a KG, we use an evidence retrieval method (such as SimGRAG (Cai et al., 2025)) for multiple relevant evidence subgraphs $\mathcal{G}_{evidence}$, where each subgraph is converted into structural text as Appendix A shows. We transform all structural texts into token sequences, and use the Marisa Trie tool to store them as a KG-Trie $\mathcal{T}$. Through this mechanism, any valid evidence subgraphs in $\mathcal{G}_{evidence}$ can be encoded into $\mathcal{T}$.

3.2.2 Token-level Constrained Decoding on KG-Trie

To ensure that each reasoning path is grounded on KG, we introduce mask-based intervention into the decoding process of LLMs based on a KG-Trie $\mathcal{T}$, which strictly confines each generated token within the token space defined by evidence subgraphs $\mathcal{G}_{evidence}$.

To adapt the LLM to KG-based FV, we first construct a task training set and obtain a KG-specialized LLM via Supervised Fine-Tuning (SFT). The format of the template in the SFT training set is shown in Appendix A, where “<path>” and “</path>” enclose the reasoning path. Each token in the reasoning path is constrained via the KG-Trie to follow valid KG paths.

In the constrained decoding mode, given the previously generated prefix context $x_{1:t-1}$ at decoding step t , we traverse the KG-Trie $\mathcal{T}$ along the token sequence $\delta(x_0, x_{1:t-1}, \mathcal{T})$ to the valid token subset: $\mathcal{V}_{valid}(x_t | x_{1:t-1}) = \{v \in \mathcal{V}_{LM} \mid \delta(x_0, x_{1:t-1}, \mathcal{T})\}$. After obtaining the raw logits z_t of the LLM, we apply a mask penalty matrix M_t to modify the output distribution as Eq. (1) shows.

$$M_t(x) = \begin{cases} 0, & \text{if } x \in \mathcal{V}_{valid}(x_{1:t-1}) \\ -\infty, & \text{otherwise} \end{cases} \quad (1)$$

The constrained probability distribution for the next token is $P(x_t | x_{1:t-1}) = \text{Softmax}(z_t + M_t)$. This mechanism forces the generation probability of any token not in $\mathcal{G}_{evidence}$ to zero during reasoning path generation. After a complete reasoning path generation under constrained decoding, LLM generates a final answer following the reasoning path without constrained decoding.

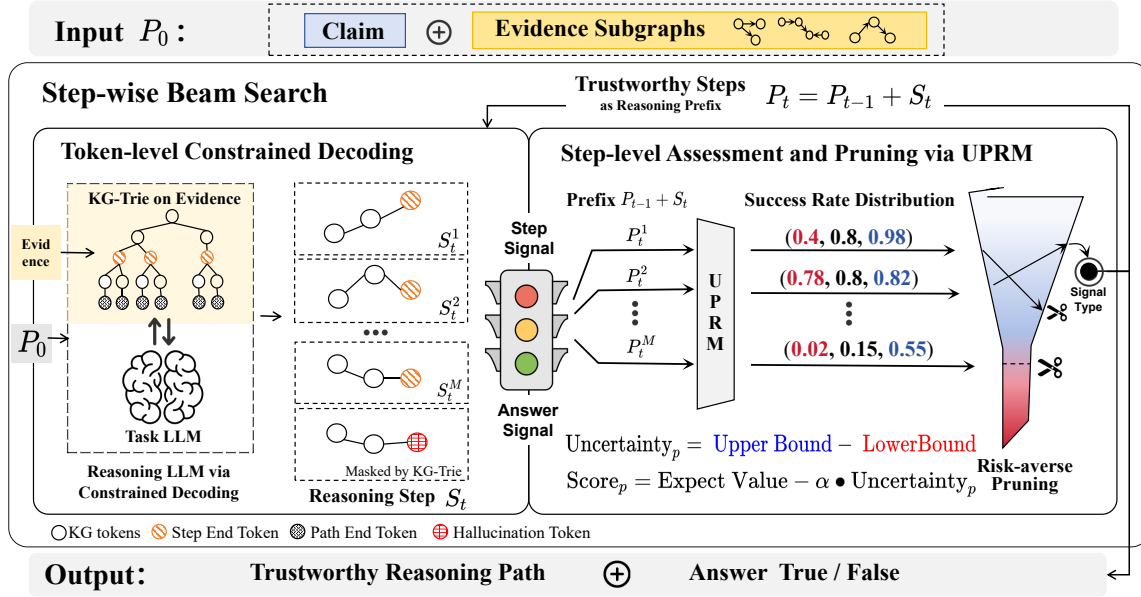


Figure 1: Overview of TrustFV Framework. TrustFV operates in three stages: (1) Token-level Constrained Decoding builds a KG-Trie for constrained decoding, eliminating structural hallucinations; (2) Uncertainty-aware Process Reward Model (UPRM) is offline-trained and outputs a discrete probability distribution over success rates; (3) Step-wise Beam Search actively prunes high-risk branches with the step-level assessment via UPRM.

3.3 Uncertainty-aware Process Reward Model

To identify plausible reasoning paths within KG, we design an Uncertainty-aware Process Reward Model (UPRM) that provides a fine-grained assessment for each step in the reasoning process. Existing Process Reward Models (PRMs) assess step quality via point estimates, which often lead to overconfident assessments due to a lack of uncertainty about the assessment (Park et al., 2025). Thus, we design a UPRM for the fine-grained step assessment with uncertainty via Monte Carlo rollouts (§3.3.1) and ordinal regression (§3.3.2).

3.3.1 Success Rate Collection via Monte Carlo Rollouts

To obtain reward signals that reflect the plausibility of each intermediate step, we collect the success rate of each reasoning step via Monte Carlo rollouts. Unlike the training set of existing PRMs that relies on expert annotations or rule-defined labels (Wang et al., 2026), we sample reasoning steps via constrained generation (§3.2), and use the success rate computed from Monte Carlo rollouts as the success rate of one step.

The specific steps are as follows. First, for each question q in the training set Q_{train} , we adopt a counterfactual sampling approach by generating reasoning paths P as seed paths with correct and incorrect answers to mitigate the distributional bias.

We decompose each seed path P into an ordered sequence of steps $\{s_1, s_2, s_3, \dots, s_N\} \in P$. For a given prefix $x_{1:t} = (s_1, s_2, \dots, s_t), t \in [1, N]$, we feed it as the initial reasoning context to the KG-specialized LLM (§3.2) and perform $N = 10$ independent Monte Carlo rollouts under KG-Trie constraints until an answer is reached. The ground-truth success rate $y_t \in [0, 1]$ for this prefix $x_{1:t}$ is defined as Eq. (2) shows:

$$y_t = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\text{rollout}_i \text{ matches target answer}), \quad (2)$$

which means the proportion of rollouts that match the golden answer. Additionally, the success rate of each golden path $x_{1:N}$ is $y_N = 1.0$ and included in the training data. Through the above procedure, we obtain a PRM training dataset, consisting of (question q , prefix $x_{1:t}$, success rate y_t).

3.3.2 Uncertainty-aware PRM via Ordinal Regression

Existing PRMs (Zheng et al., 2025) only capture relative step quality without quantifying the confidence of their assessments, which can cause them to favor high-scoring yet uncertain steps over sub-optimal but reliable ones, leading to error cascading. To address this, we design an Uncertainty-aware PRM (UPRM) via ordinal regression. Instead of regressing on a continuous success rate,

which cannot express assessment uncertainty, we discretize the continuous success rate into ordered intervals and train the UPRM to predict a full probability distribution over them, jointly capturing expected success and predictive uncertainty.

Success Rate Discretization and Label Construction. We uniformly discretize the target interval $[0, 1]$ into M equal-width ordinal bins, yielding an ordered category set $\mathcal{B} = \{B_0, B_1, \dots, B_{M-1}\}$. B_m covers the probability range $[\frac{m}{M}, \frac{m+1}{M})$, with the last bin B_{M-1} taken as $[\frac{M-1}{M}, 1]$ to include the right endpoint. The discrete label for a continuous success rate y_t is the unique bin B_m whose index m is given by Eq. (3):

$$m = \min(\lfloor y_t \times M \rfloor, M - 1). \quad (3)$$

UPRM Architecture. We build upon the KG-specialized LLM (§3.2) and replace its language modeling head with a linear layer $\mathbf{W}_{PRM} \in \mathbb{R}^{d \times M}$ that projects to M output dimensions. For a given step prefix $p_{0:t}$, we append a special token (eg, `<extra_0>`) at its end and take the final-layer hidden state $\mathbf{h}_t \in \mathbb{R}^d$ corresponding to this token as the prefix representation. The logits are then computed as $\mathbf{z}_t = \mathbf{W}_{PRM} \mathbf{h}_t \in \mathbb{R}^M$, and a softmax function yields the predicted probability distribution over the M success rate bins:

$$P(B_m | x_{1:t}) = \frac{\exp(z_{t,m})}{\sum_{j=1}^M \exp(z_{t,j})}, \quad (4)$$

where $B_m, m \in (0, \dots, M - 1)$ denotes the m -th success rate bin and $z_{t,m}$ is the corresponding logit.

Distance-Weighted Ordinal Regression Loss. Since the success rate labels are inherently ordered, we formulate the learning as an ordinal regression task and design a distance-weighted cross-entropy loss. This loss explicitly penalizes misclassifications in proportion to the absolute ordinal distance $|m - c_t|$, forcing the model to respect the ordered structure of the bins. For a prefix $x_{1:t}$ with ground-truth bin index $c_t \in \{0, 1, \dots, M - 1\}$ and predicted probability distribution $p_m = P_\theta(B_m | x_{1:t})$, the loss is defined as:

$$\mathcal{L}(c_t, x_{1:t}) = -\log p_{c_t} - \lambda \sum_{m \neq c_t} |m - c_t| \log(1 - p_m), \quad (5)$$

where λ balances the primary classification term and the ordinal penalty. The first term maximizes the probability of the correct bin, while the second term suppresses incorrect predictions with a strength proportional to their distance from the

truth, thereby enabling the UPRM to learn a well-calibrated ordered distribution.

3.4 Step-wise Beam Search for Trustworthy Reasoning

To generate trustworthy reasoning paths in KG-based FV, we design a step-wise beam search strategy that leverages UPRM to evaluate each reasoning step and proactively prune high-risk branches. Existing methods (Luo et al., 2025) rely on LLMs’ raw logits for branch selection, which tend to lock in error branches due to inherent knowledge gaps. To select plausible branches more reliably, our approach prunes high-risk branches with fine-grained step assessment. Concretely, we first design a risk-averse reasoning step assessment via UPRM (§3.4.1), and then use these step-level signals to prune high-risk branches (§3.4.2) in beam search. This ensures that the reasoning paths remain focused on plausible steps, ultimately producing a trustworthy reasoning path and a final answer.

3.4.1 Step Risk Assessment via UPRM

To accurately estimate the risk of a reasoning step deviating from a plausible path, we use the full success probability distribution of this step predicted by the UPRM to quantify the step score. Existing PRM-based path evaluation methods predominantly rely on a single score to select preferred paths (Wang et al., 2026) without confidence, which are easily misled by erroneous branches that appear valid yet lead to an incorrect answer. By extracting the expected success probability and the prediction spread from the predicted distribution, we achieve fine-grained step assessment consistent of success probability and its uncertainty.

For each reasoning step s produced by the KG-specialized LLM (§3.2) through step-wise constrained decoding over the KG-Trie, we concatenate s_t with all preceding steps to form a reasoning prefix $x_{1:t}$. Taking this prefix as input, UPRM predict the probability distribution over M success rate bins. From this distribution, we compute the expected success rate $\mathbb{E}[s_t | x_{1:t}]$ as Eq. (6) shows:

$$\mathbb{E}[s_t | x_{1:t}] = \sum_{m=1}^M P(B_m | x_{1:t}) \cdot v_m, \quad (6)$$

where $P(B_m | x_{1:t})$ is the predicted probability that prefix $x_{1:t}$ falls into the bin B_m , and v_m is the median value of the success rate interval of B_m .

Unlike prior work (Park et al., 2025) that only uses the 10th percentiles as uncertainty, we define uncertainty ΔU as the probability gap between the pessimistic lower bound (10th percentile) and the optimistic upper bound (90th percentile) of the predicted distribution. This gap directly measures the width of probability distribution. When the predicted probability mass is sharply concentrated in one or two adjacent bins, the 10th and 90th percentiles lie close together and ΔU is small, reflecting low uncertainty. Otherwise, the large ΔU means high uncertainty.

To compute ΔU , we use the full probability distribution produced by UPRM, we first compute the cumulative probability over bins $F_m = \sum_{j=1}^m P(B_j|x_{0:t})$. For a quantile level $q \in [0, 1]$, the corresponding bin index is defined as: $m(q) = \min\{m \in \{1, \dots, M\} \mid F_m \geq q\}$. The uncertainty is defined as $\Delta U(x_{1:t}) = v_{m(0.9)} - v_{m(0.1)}$, where $v_{m(q)}$ means the median value of the success rate interval represented by bin $m(q)$. A wider ΔU indicates a more dispersed predictive distribution, signifying greater disagreement and higher uncertainty in the current steps assessment.

Finally, we define the risk score of one step as a linear combination of the expected success rate and the prediction spread:

$$\text{Score}(x_{1:t}) = \mathbb{E}[S|x_{1:t}] - \alpha \cdot \Delta U(x_{1:t}), \quad (7)$$

where α is a tunable penalty coefficient. This scoring penalizes highly uncertain steps, guiding beam search to favor paths with both high expected success and low uncertainty, thereby suppressing hallucinations at the source.

3.4.2 Step-wise Beam Search via Risk-averse Pruning

To focus exploration on plausible branches, we perform a step-wise beam search over the KG-Trie guided by step risk assessment for pruning high-risk paths with high uncertainty proactively.

With beam size B , each iteration proceeds in step-wise expansion and risk-averse pruning, until answer generation. In step-wise expansion, for each active branches $x_{1:t-1} = (s_1, \dots, s_{t-1})$, the LLM generates the next reasoning step s_t under constrained decoding (§3.2) and removes duplication as a candidate set $X_{1:t}$. In risk-averse pruning, candidate branches with expected success rate $\mathbb{E}(x|x_{1:t} \in X_{1:t}) < 0.5$ are filtered out, which are highly likely to lead to a wrong answer with below-chance success probability. The remaining

candidates are ranked by the score $\text{Score}(x_{1:t})$ (Eq. 7), and only the top- P branches are retained for the next iteration, which have relatively high expected success and low uncertainty. Finally, until all active branches emit the termination token “<answer>”. The path with the highest scores and its answer are selected as the final verification conclusion, while the plausible reasoning path serves as the trustworthy verification process.

4 Experiments

4.1 Experiment Setups

Datasets. Following previous works, we conduct experiments on two datasets, the FactKG dataset and the Complex WebQuestions (CWQ). The detailed statistics of the datasets in Appendix B.

Baselines. We compare TrustFV with 16 baselines grouped into two classifications: 1) GNNs-based methods: GEAR (Zhou et al., 2019), MHGCI (Chen et al., 2025), NSM (He et al., 2021), and UniKGQA (Jiang et al., 2023); 2) LLM-based Reasoning methods: ChatGPT, Llama3-70B, Llama3-8B, KG-GPT (Kim et al., 2023a), ToG (Sun et al., 2024), RoG (Luo et al., 2024), SimGRAG (Cai et al., 2025), xRAG (Cheng et al., 2024), DPRM (Wang et al., 2026), PGR (Hao and Wu, 2025), GCR (Luo et al., 2025), and SIE (Yu et al., 2026). The detailed baselines in Appendix D.

Evaluation Metrics. We adopt Accuracy and Hit@1 as the evaluation metrics following previous work (Cai et al., 2025) on FactKG and CWQ. Accuracy considers the correctness of the verification answer, while Hit@1 checks whether the generated answer is among the correct answers.

Our implementation details and hyperparameter settings are listed in Appendix C.

4.2 Main Results

As Tab.1 shows, TrustFV consistently achieves SOTA performance on two datasets, which are reported with standard deviation over five runs. On FactKG, TrustFV attains 90.07(± 1.6)% accuracy, outperforming the prior best baseline PGR by a clear margin (+3.25%). On the QA dataset CWQ, TrustFV reaches 87.62% Hit@1, outperforming the latest baseline SIE by 1.9%. These results validate the effectiveness of our framework for LLM reasoning on KG. Compared to the KG-Trie-based decoding method GCR, TrustFV yields improvements of 4.77% on FactKG and 11.1% on CWQ, showing that process reward-guided step-wise pruning

effectively tackles the error cascading.

We further perform rigorous statistical significance tests against the strongest baselines to confirm the reliability of our performance gains. All improvements are statistically significant ($p < 0.001$); detailed test settings and complete results are provided in Appendix F.

Model	FactKG (Acc %)	CWQ (Hit@1 %)
<i>GNNs-based Methods</i>		
GEAR (Zhou et al., 2019)	78.58	–
MHGCI (Chen et al., 2025)	84.41	–
NSM (He et al., 2021)	–	47.6
UniKGQA (Jiang et al., 2023)	–	51.2
<i>LLM-based Reasoning Methods</i>		
ChatGPT	68.50	34.70
Llama3-70B	68.40	35.20
Llama3-8B	56.92	28.10
KG-GPT (Kim et al., 2023a)	71.00	–
ToG (GPT-4) (Sun et al., 2024)	–	68.50
RoG (Llama2-7B) (Luo et al., 2024)	–	62.60
SimGRAG (Llama3-70B) (Cai et al., 2025)	86.80	70.50
xRAG (Mixtral-8x7B) (Cheng et al., 2024)	70.42	–
DPRM (Llama3.1-8B) (Wang et al., 2026)	74.88	72.11
PGR (GPT-4o-mini) (Hao and Wu, 2025)	<u>86.82</u>	–
GCR (Llama3.1-8B+GPT-4o) (Luo et al., 2025)	85.30	76.52
SIE w/RL (Llama3.1-8B) (Yu et al., 2026)	86.43	<u>85.72</u>
TrustFV(Ours)	90.07(±1.6)	87.62(±1.7)

Table 1: Overall performance comparison on FactKG and CWQ. The best results are highlighted in **bold**, and the second-best are underlined. Blank cells indicate that the results are not reported in the original papers.

4.3 Ablation Studies

We conduct ablation studies on both FactKG and CWQ datasets to quantify the contribution of each core component of TrustFV, with all results averaged over 5 independent runs (Table 2). We detail each variant and corresponding findings as follows.

w/o Token-level Constrained Decoding removes the constraint decoding mechanism, allowing the LLM to generate tokens freely. Removing this module causes the sharpest performance drop (23.34% on FactKG, 31.62% on CWQ). This confirms that hard KG-based constraints are the foundational guarantee for eliminating structural hallucinations and ensuring all reasoning paths are grounded in the KG.

w/o Uncertainty-aware PRM replaces UPRM step assessment with raw LLM logits for branch ranking, discarding both success probability estimation and uncertainty modeling. The substantial performance decline (22.03% on FactKG, 12.62% on CWQ) demonstrates that constrained decoding alone is insufficient to select high-quality paths, and fine-grained step-level reward evaluation is critical for guiding reliable reasoning.

w/o Distance-Weighted Loss Function trains the UPRM with a standard cross-entropy loss function over ordinal bins instead of the proposed

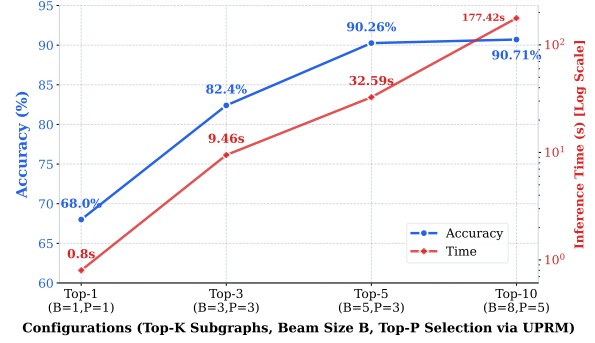


Figure 2: Sensitivity Analysis of Key Parameters on FactKG over 3 Runs. Accuracy (left y-axis) and inference time (right y-axis) are reported for each configuration.

distance-weighted ordinal regression loss, with all other settings unchanged. The notable performance drop (3.58% on FactKG, 4.15% on CWQ) validates that the distance-weighted penalty effectively exploits the ordered nature of success rate labels, improving step ranking accuracy and benefiting downstream pruning.

w/o Step-wise Beam Search removes intermediate step-level pruning and operates as path-level reranking: it first generates all complete KG-constrained reasoning paths under the same compute budget, then applies UPRM scoring only at the final path level. Performance drops by 15.84% on FactKG and 10.5% on CWQ, indicating that most performance gains come from proactive step-wise pruning that suppresses error cascading, rather than from UPRM path scoring alone.

All four components contribute consistently to the final performance. Their synergy enables TrustFV to generate faithful, plausible reasoning paths for trustworthy fact verification.

Ablation Setting	FactKG (Acc %)	CWQ (Hit@1 %)
Ours	90.07 (±1.6)	87.62 (±1.7)
w/o Token-level Constrained Decoding	66.73 (±0.8)	56.00 (±1.1)
w/o Uncertainty-aware PRM	68.04 (±0.5)	75.00 (±0.8)
w/o Distance-Weighted Loss Function	86.49 (±2.21)	83.47 (±0.3)
w/o Step-wise Beam Search	74.23 (±1.2)	77.12 (±1.4)

Table 2: Ablation study of TrustFV on FactKG and CWQ datasets. Removing any core component leads to a drop, validating the necessity of each component.

4.4 Sensitivity Analysis of Key Parameters

To study parameter sensitivity, we vary the number of evidence subgraphs K , beam size B , and retained candidates size P after pruning on FactKG.

Fig. 2 reports the accuracy and inference time under the optimal (B, P) combination for each K ,

with full results provided in Appendix E. As K grows from 1 to 5, accuracy rises sharply from 68.0% to 90.26%, confirming that moderate evidence expansion is critical for reasoning quality. Beyond $K = 5$, however, accuracy nearly plateaus, indicating rapidly diminishing returns from additional evidence. In stark contrast, $K = 10$ costs over four times as much as $K = 5$, and the inference time grows exponentially, indicating that larger K demands a wider beam search space and causes a surge in candidate branches. These results suggest that $K = 5$ achieves the optimal accuracy with efficiency trade-off, beyond which marginal precision gains come at high computational cost.

4.5 Sensitivity Analysis of UPRM Parameter

To evaluate the ability of UPRM to discern reasoning path quality, we sample 1,000 positive and negative paths generated by KG-specialized LLM, score them with UPRM under varying the risk parameter α in Eq.(7), and report both task performance and path discrimination capability (AUROC). Results are shown in Tab. 3.

UPRM with $\alpha = 0.2$ achieves the best performance across all metrics: 90.07% Acc and 70.13% AUROC on FactKG, 87.62% Hit@1 and 77.94% AUROC on CWQ. This moderate risk penalty effectively balances pursuing high expected success against avoiding high uncertainty. When $\alpha = 0.0$, the model relies solely on expected success rate while ignoring uncertainty, leading to clear performance drops (FactKG Acc: 85.29%; AUROC: 67.86%), confirming the critical value of uncertainty information for path quality assessment. Increasing α to 0.5 induces excessive risk aversion, causing performance to deteriorate sharply and even fall, indicating that overly heavy uncertainty penalties mistakenly discard valuable paths.

Table 3: Sensitivity Analysis of UPRM Parameter (mean $\pm$ std over 5 runs).

Method	FactKG		CWQ	
	Acc (%)	AUROC	Hit@1 (%)	AUROC
UPRM ($\alpha=0.0$)	85.29 (± 1.8)	67.86 (± 0.7)	82.26 (± 1.3)	71.62 (± 0.7)
UPRM ($\alpha=0.5$)	72.51 (± 1.5)	58.70 (± 1.0)	78.30 (± 1.3)	66.04 (± 0.9)
UPRM ($\alpha=0.2$)	90.07 (± 1.6)	70.13 (± 0.5)	87.62 (± 1.7)	77.94 (± 0.5)

4.6 Inference Time Analysis

To gain deeper insight into the performance of TrustFV across varying complex problems, we decompose accuracy and average inference time by question type, as reported in Tab. 3.

TrustFV proves both efficient and accurate on simpler questions: on FactKG Existence and 1-hop categories, it achieves 96% and 89.13% accuracy with average inference times of merely 14.09 s and 10.55 s, respectively. For more challenging multi-hop reasoning (FactKG multi-hop and CWQ 3-hop), the model reasonably expends more computation (39.19 s and 36.51 s) while still maintaining strong performance at 88.14% and 83.33%, demonstrating the effectiveness of the active pruning mechanism during beam search in KG-Tries. From CWQ 2-hop to 3-hop, accuracy drops 4.72 % with a time increase of 39%, indicating that TrustFV scales well to deeper reasoning without catastrophic degradation. Moreover, on multi-claim and negation categories that require cross-evidence validation and logical negation, the model performs robustly (87.12% and 88.63%), confirming its broad adaptability to diverse reasoning patterns. Taken together, TrustFV adaptively allocates computation according to problem complexity, achieving a consistently favorable trade-off between efficiency and accuracy.

Table 4: Performance on Different FactKG Subtasks (averaged over 5 runs).

Metric	Existence	1-hop	Multi-hop	Multi-claim	Negation
Acc (%)	96.00	89.13	88.14	87.12	88.63
Time (s)	14.09	10.55	39.19	25.60	15.30

Table 5: Performance on CWQ with Different Hops (averaged over 5 runs).

Metric	1-hop	2-hop	3-hop
Hit@1 (%)	91.60	88.05	83.33
Time (s)	10.80	26.21	36.51

4.7 Reasoning Trustworthiness Analysis

To evaluate the trustworthiness of the generated paths, we compare the performance of TrustFV with UPRM and with a logits-based pruning strategy on top of KG-Tries decoding. We set two dimensions: (1) faithfulness to the KG, measured by whether all triples exist in the KG; and (2) alignment with gold evidence, measured by whether the generated path matches the annotated golden path. Fig. 3 shows that on top of the constrained decoding mechanism, all paths are 100% grounded in the KG for eliminating structural hallucinations. Furthermore, the alignment with golden paths increases from 84% to 92% on FactKG and from

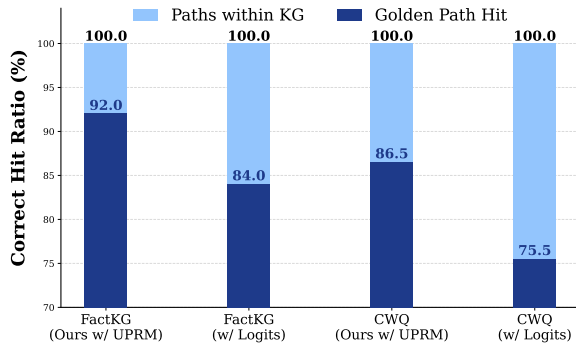


Figure 3: Reasoning trustworthiness analysis under different pruning strategies on two datasets. Two metrics are evaluated: Paths within KG and Golden Path Hit (alignment with gold evidence), averaged over 5 runs.

75.5% to 86.5% on CWQ, demonstrating that risk-averse step-wise pruning more effectively identifies plausible reasoning paths. In summary, UPRM significantly enhances path plausibility while constrained decoding ensures paths within KG.

5 Conclusion

In this paper, we propose TrustFV, a framework to mitigate the error cascading issue in KG-based fact verification. TrustFV aims to prune high-risk branches during constrained decoding on KG-Tries for a step-wise beam search, thereby generating plausible reasoning paths without structural hallucination for a trustworthy reasoning process. We explicitly model step-level uncertainty and move the pruning stage from the path level to the step level, fundamentally for tackling error cascading. Experiments on FactKG and CWQ show that TrustFV achieves SOTA performance and generates reasoning paths superior in faithfulness and plausibility.

Limitations

Despite the strong performance of TrustFV on some KG-based tasks, one aspect that merits further improvement is its fixed beam width and pruning strategy, whereas the required search space varies considerably across claims of different complexity. Future work could adaptively adjust the search width based on UPRM’s uncertainty scores, achieving a better balance between efficiency and accuracy.

Ethical statement

This research focuses on providing trustworthy fact verification techniques for misinformation detec-

tion, offering positive social value. The experiments use publicly available academic datasets, FactKG and CWQ, which contain no personally identifiable information.

6 Acknowledgments

This work was supported by the following funding sources:

1. The National Key Research and Development Program of China (Grant No. 2023YFC3306100).
2. The Innovation Research Foundation of National University of Defense Technology (Grant No. ZK25-12).

And we would like to express sincere gratitude to the anonymous reviewers for their valuable feedback.

References

- Giannis Bekoulis, Christina Papagiannopoulou, and Nikos Deligiannis. 2023. [A review on fact extraction and verification](#). *ACM Comput. Surv.*, 55(2):12:1–12:35.
- Yuzheng Cai, Zhenyue Guo, Yiwen Pei, Wanrui Bian, and Weiguo Zheng. 2025. [Simgrag: Leveraging similar subgraphs for knowledge graphs driven retrieval-augmented generation](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 3139–3158. Association for Computational Linguistics.
- Daoqi Chen, Yaxin Li, Zizhong Zhu, Xiaowang Zhang, and Zhiyong Feng. 2025. [Faithful inference chains extraction for fact verification over multi-view heterogeneous graph with causal intervention](#). In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 4634–4645. Association for Computational Linguistics.
- Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. [xrag: Extreme context compression for retrieval-augmented generation with one token](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Yuanzhen Hao and Desheng Wu. 2025. [Fact verification on knowledge graph via programmatic graph reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 5480–5495. Association for Computational Linguistics.

- Gaole He, Yunshi Lan, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. [Improving multi-hop knowledge base question answering by learning intermediate supervision signals](#). In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pages 553–561. ACM.
- Jinhao Jiang, Kun Zhou, Xin Zhao, and Ji-Rong Wen. 2023. [Unikgqa: Unified retrieval and reasoning for solving multi-hop question answering over knowledge graph](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jiho Kim, Yeonsu Kwon, Yohan Jo, and Edward Choi. 2023a. [KG-GPT: A general framework for reasoning on knowledge graphs using large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, Findings of ACL, pages 9410–9421. Association for Computational Linguistics.
- Jiho Kim, Sungjin Park, Yeonsu Kwon, Yohan Jo, James Thorne, and Edward Choi. 2023b. [Factkg: Fact verification via reasoning on knowledge graphs](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16190–16206. Association for Computational Linguistics.
- Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheon-Eum Park, and Kyomin Jung. 2021. [Crossaug: A contrastive data augmentation method for debiasing fact verification models](#). In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3181–3185. ACM.
- Kun Li, Tianhua Zhang, Xixin Wu, Hongyin Luo, James R. Glass, and Helen M. Meng. 2025. [Decoding on graphs: Faithful and sound reasoning on knowledge graphs through generation of well-formed chains](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 24349–24364. Association for Computational Linguistics.
- Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. [Reasoning on graphs: Faithful and interpretable large language model reasoning](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Yuan-Fang Li, Chen Gong, and Shirui Pan. 2025. [Graph-constrained reasoning: Faithful reasoning on knowledge graphs with large language models](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.
- Young-Jin Park, Kristjan Greenewald, Kaveh Alimohammadi, Hao Wang, and Navid Azizan. 2025. [Know what you don't know: Uncertainty calibration of process reward models](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 38852–38895. Curran Associates, Inc.
- Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. [Get your vitamin c! robust fact verification with contrastive evidence](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 624–643. Association for Computational Linguistics.
- Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel M. Ni, Heung-Yeung Shum, and Jian Guo. 2024. [Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Sheng Tian, Xintan Zeng, Yifei Hu, Baokun Wang, Yongchao Liu, Yue Jin, Changhua Meng, Chuntao Hong, Tianyi Zhang, and Weiqiang Wang. 2024. [Graphrpm: Risk pattern mining on industrial large attributed graphs](#). In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track - European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9-13, 2024, Proceedings, Part X, Lecture Notes in Computer Science*, pages 133–149. Springer.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. [Math-shepherd: Verify and reinforce llms step-by-step without human annotations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.
- Xinyi Wang, Yiping Song, Zhiliang Tian, Bo Liu, Tingjin Luo, and Minlie Huang. 2026. [DPRM: A dual implicit process reward model in multi-hop question answering](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 33683–33691. AAAI Press.
- Zhangyue Yin, Qiushi Sun, Zhiyuan Zeng, Qinyuan Cheng, Xipeng Qiu, and Xuanjing Huang. 2025. [Dynamic and generalizable process reward modeling](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 4203–4233. Association for Computational Linguistics.

Peng Yu, Zeyuan Zhao, Shao Zhang, Luoyi Fu, Xinbing Wang, and Ying Wen. 2026. [Structured in-context environment scaling for large language model reasoning](#). In *The Fourteenth International Conference on Learning Representations*.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2026. [Genprm: Scaling test-time compute of process reward models via generative reasoning](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 34932–34940. AAAI Press.

Congming Zheng, Jiachen Zhu, Zhuoying Ou, Yuxiang Chen, Kangning Zhang, Rong Shan, Zeyu Zheng, Mengyue Yang, Jianghao Lin, Yong Yu, and Weinan Zhang. 2025. [A survey of process reward models: From outcome signals to process supervisions for large language models](#). *CoRR*, abs/2510.08049.

Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. [GEAR: graph-based evidence aggregating and reasoning for fact verification](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 892–901. Association for Computational Linguistics.

A Appendix A: Prompt Template

In this section, we illustrate all the prompt templates used in TrustFV.

Subgraph Sentence Template. As Fig. 4 and Fig. 5 show, the template for converting subgraphs retrieved from KG into natural language sentences.

Reasoning Paths on Fact Verification

```
<path>(Entity 1, Relation 1, Entity
2);(Entity 3, Relation 2, Entity 4).
Relation: Support/ Irrelevant</path>
```

Figure 4: The structural text of an Evidence Subgraph on FV

Reasoning Paths on KGQA

```
<path>(Entity 1, Relation 1, Entity
2);(Entity 2, Relation 2, Entity 3)</path>
```

Figure 5: The structural text of Reasoning Path on KGQA

SFT Template for FV and KGQA. As Fig. 6 and Fig. 7 show, the claim and retrieved evidence

subgraphs are provided to the LLMs to generate a reasoning path and an answer. As Fig. 8 and Fig. 9 show, the question and retrieved KG paths are provided to the LLMs to generate a reasoning path and an answer.

FV Prompt Template

You are a precise knowledge graph reasoning engine. You must find the relevant evidence graphs for the claim, and output the (Support, Irrelevant) for each evidence graph and claim strictly enclosed in <path> </path>.

Support means that the facts of this evidence graph are consistent with the facts of the claim, and this claim is true. Answer: True. Otherwise, answer: False. "Answer: True" means that the facts of this evidence graph are consistent with the facts of the claim, and this claim is true. Otherwise, evidence is irrelevant to the claim.

Example:

A multi-hop claim: School is the final destination from the Mall via x1 and x2.

Available evidence graphs:

evidence 1: <path> (Mall,x1,Railway station);(Railway station,x3,School) </path>

evidence 2: <path> (Mall,x1,Park);(Park,x2,Hospital) </path>

Output:

<path>(Mall,x1,Park);(Park,x2,Hospital).

Relation: Irrelevant </path>

<answer> False </answer>

An existence claim: What is the name of M. K. J. Cantlay’s award?

Available evidence graphs:

evidence 1: <path>(M. K. J. Cantlay,award,Efficiency Decoration) </path>

evidence 2: <path>(M. K. M,award,Nober) </path>

evidence 3: <path>(M. T. J,award,JinJiPrice) </path>

Output:

<path>(M. K. J. Cantlay,award,Efficiency Decoration).Relation: Support </path>

<answer> True </answer>

A {Type} claim: {claim}

Available evidence graphs:{paths}

Output:

Figure 6: FV Prompt Template

B Appendix B: Datasets

We conduct experiments on two fact verification datasets: FactKG and CWQ. FactKG consists of 99,133 training, 500 validation, and 9,041 test samples, with 4,398 true and 4,643 false claims in the

FV Answer Template

```
<path>{golden paths}. Relation:
{relation}</path> <answer> {golden answers}
</answer>
```

Figure 7: FV Answer Prompt Template

	Train	Evaluate	Test
FactKG	99,133	500	9041
CWQ	27139	500	3531

Table 6: Statistics of datasets.

test set. The test set covers diverse claim types: 1914 single-hop, 3069 conjunction, 870 existence, 1874 multi-hop, and 1314 negation. CWQ contains 27639 training and 3531 test samples in total, with question hop counts ranging from 1 to 4, and we sample 500 evaluation samples from the training samples.

KGQA Prompt Template

You are an accurate knowledge graph reasoning engine. You must strictly output the reasoning path in `<path>` `</path>`, and then output the final answer. If there are multiple answers, output the corresponding path and an answer for each, with each path and answer on a separate line.

Example:
Query: What is the final destination from Aplion via x1 and x2?
Available paths:
Path 1: `<path>(Aplion,x1,Blion);(Blion,x2,Clion)</path>`
Path 2: `<path>(Aplion,x3,Mik),(Mik,x2,Taxi)</path>`
Output:
`<path>(Aplion,x1,Blion);(Blion,x2,Clion)</path>`
`<answer> Clion </answer>`
Query: {query}
Available paths: {paths}
Output:

Figure 8: KGQA Prompt Template

KGQA Prompt Template

```
<path>{golden path}</paths> <answer>
{answer entity} </answer>
```

Figure 9: KGQA Answer Template

C Appendix C: Implementation Settings

We train a KG-specialized LLM based on Llama-3.1-8B-Instruct using LoRA for 1,000 steps, and further train UPRM on top of this KG-specialized LLM. For UPRM training, we use the AdamW optimizer with a learning rate of 5×10^{-5} , a batch size of 4, and train for 50 epochs. The number of ordinal bins is set to 5, with the distance weight exponent set to 1. During inference, we empirically set the beam size to 5, use 5 evidence subgraphs, and retain the top-3 steps after pruning. All experiments are performed on two NVIDIA A800 80GB PCIe GPUs.

D Appendix D: Baselines

We compare TrustFV with 16 baselines grouped into two categories: (1) GNNs-based Methods, (2) LLMs Reasoning-based Methods. The details of each baseline are described as follows.

GNNs-based Methods

GEAR(Zhou et al., 2019): A framework that transfers information across a fully-connected evidence graph and utilizes permutation-invariant aggregators with BERT representations to evaluate multi-evidence fact verification.

MHGCI(Chen et al., 2025): A self-optimizing causal intervention approach that extracts faithful inference chains while mitigating out-of-distribution correlation issues through invariant risk minimization and backdoor adjustment.

NSM(He et al., 2021): A bidirectional reasoning architecture that trains a student navigation network using dense intermediate supervision signals derived from a teacher network’s simultaneous forward and backward graph traversals.

UniKGQA(Jiang et al., 2023): A unified retrieval and reasoning model that bridges semantic gaps through shared contrastive pre-training and specialized fine-tuning on abstract subgraphs to eliminate multi-stage cascading errors.

LLMs Reasoning-Based Methods

ChatGPT: A foundational large language model evaluated in zero-shot and few-shot settings to es-

tablish a baseline for parametric knowledge retrieval, alignment, and pure natural language inference.

Llama3-70B: A massive open-source dense language model utilized to benchmark the absolute upper theoretical bounds of ungrounded, heavily parameterized memory-based reasoning capabilities.

Llama3.1-8B: A lightweight, highly optimized open-source language model serving as a performance floor for efficient, localized parametric reasoning without external structural augmentation.

KG-GPT(Kim et al., 2023a): A cohesive, modular framework that segments complex queries, retrieves specific graph components, and prompts language models to derive logical conclusions directly over structured data.

ToG(Sun et al., 2024): An exploration strategy that treats the language model as an autonomous agent, iterating through a deep, responsible think-search-generate loop to actively prune and traverse knowledge graphs.

RoG(Luo et al., 2024): A strict planning-retrieval-reasoning paradigm that prompts language models to generate physically grounded relation paths as plans, enabling interpretable reasoning on retrieved valid subgraphs.

SimGRAG(Cai et al., 2025): A retrieval-augmented generation method that optimizes semantic alignment by matching abstract query patterns to isomorphic real-world knowledge subgraphs utilizing Graph Semantic Distance.

xRAG(Cheng et al., 2024): An extreme context compression technique that condenses vast retrieved structural evidence into a single virtual token via modality fusion to drastically reduce computational overhead and FLOPs.

DPRM(Wang et al., 2026): A reinforcement learning approach that trains dual implicit process reward models without manual annotations to mutually verify and penalize chain-of-thought texts against knowledge graph paths.

PGR(Hao and Wu, 2025): A neuro-symbolic method that enforces absolute logical rigor in fact verification by converting implicit neural entailment predictions into explicit, deterministic, executable programmatic operations over graphs.

GCR(Luo et al., 2025): A zero-hallucination architecture that hard-constrains language model autoregressive decoding using a KG-Trie mask while leveraging dual models for constant-time path traversal and inductive consensus.

SIE(Yu et al., 2026): A progressive reinforcement learning methodology that automatically extracts structured in-context environments from graphs with varying retention ratios to scale and generalize the logical capabilities of LLMs.

E Appendix E: Parameter Analysis Experiment

We analyze the impact of evidence subgraph count K , beam size B , and selection size P on the FactKG dataset. The accuracy and inference latency under different parameter settings are in Tab. 9.

Effect of Subgraph Count K (1) Under a small beam size ($B = 1, P = 1$), increasing K yields limited gains. Without beam search or pruning, accuracy rises only from 68.0% to 78.0% as K grows from 1 to 10, yet still far below configurations with active search. This indicates that merely stacking subgraphs without effective search and pruning cannot fully exploit the evidence.

(2) Under a moderate beam size ($B = 3, P = 3$), K brings more evident improvements. Accuracy increases from 82.40% to 88.14% (+5.74 %) when K goes from 3 to 10, confirming that with sufficient search capability, richer evidence can be effectively utilized. However, inference time also rises from 9.46 s to 26.62 s, nearly a threefold increase.

(3) Under a large beam size ($B = 5, P = 5$), K continues to deliver gains but with clear efficiency degradation. Accuracy edges up from 88.63% to 89.88% (+1.25%) when K increases from 5 to 10, while latency jumps from 29.60 s to 52.73 s (+78%), further corroborating the general trend of diminishing accuracy returns and rapidly growing computational cost.

Effect of Beam Size B and Selection Size P (1) Search size is critical but should be moderate. At $K = 5$, increasing (B, P) from (1,0) to (3,3) boosts accuracy from 74.23% to 87.05% (+12.82%), demonstrating the necessity of a reasonable beam size for exploring high-quality paths. Further increasing to (5,3) achieves the peak of 90.26%, but extending to (5,5) surprisingly drops accuracy to 88.63%. This inverted-U pattern suggests that an overly large P may aggressively prune away valuable candidates.

(2) P and B must be set in coordination. Comparing configurations at $K = 10$: ($B=5, P=3$) attains 90.72%, ($B=8, P=3$) yields 90.50%, and ($B=8, P=5$) reaches 90.71%. Simply widening the beam does not monotonically improve perfor-

mance, yet it dramatically inflates latency (50.30 s, 108.25 s, and 177.42 s, respectively). This indicates that proactive pruning already effectively identifies high-quality paths under moderate beam sizes, while excessive search introduces noise.

In summary, the subgraph count K provides the essential evidence foundation, and an overly small K severely limits coverage. Beam search and pruning are the core drivers of performance improvements; a moderate increase in beam size can yield double-digit accuracy gains. Proactive pruning effectively curbs the explosion of search cost; a suitable P maintains high accuracy while significantly controlling latency. The configuration ($K = 5, B = 3, P = 5$) achieves the best trade-off with 90.26% accuracy in 32.59 seconds, while ($K = 10, B = 5, P = 3$) is a competitive alternative at 90.72% accuracy with 50.30 seconds.

Beam Size B Top-P Selection	Top-K Evidence			
	K=1	K=3	K=5	K=10
B=1,P=1	68.00	72.51	74.23	78.00
B=3,P=3	–	82.40	87.05	88.14
B=5,P=3	–	–	90.26	90.72
B=5,P=5	–	–	88.63	89.88
B=8,P=3	–	–	–	90.50
B=8,P=5	–	–	–	90.71

Table 7: **Parameter Sensitivity (Acc %)**

Beam Size B Top-P Selection	Top-K Evidence			
	K=1	K=3	K=5	K=10
B=1,P=1	0.80	1.16	1.21	1.43
B=3,P=3	–	9.46	14.55	26.62
B=5,P=3	–	–	32.59	50.30
B=5,P=5	–	–	29.60	52.73
B=8,P=3	–	–	–	108.25
B=8,P=5	–	–	–	177.42

Table 8: **Inference Time (s)**

Table 9: Parameter sensitivity analysis of beam size B , prune number P , and top- K evidence.

F Appendix F: Statistical Significance Tests

F.1 Experimental Setup

To verify that the performance improvements of TrustFV are not caused by random variance, we reproduce the strongest baseline on each dataset

under exactly the same experimental settings, and conduct 5 independent runs for both baselines and our method for fair comparison. We adopt three complementary statistical tests for comprehensive verification:

1. **Paired t-test:** tests the significance of the difference in mean performance across repeated runs.
2. **Wilcoxon signed-rank test:** a non-parametric test that does not assume a normal distribution of results, serving as a robust alternative to the t-test.
3. **Sample-level binomial test:** we count the win/tie/loss distribution at the individual sample level, and use a binomial test to verify the statistical significance of our method’s sample-level superiority.

F.2 Results on FactKG

We compare TrustFV with the strongest baseline PGR on the FactKG dataset:

- PGR: $85.13 \pm 2.9\%$ accuracy
- TrustFV (Ours): $90.07 \pm 1.6\%$ accuracy

The 4.94% absolute accuracy improvement is statistically significant under both the paired t-test and the Wilcoxon signed-rank test (all $p < 0.001$), confirming that the performance advantage of our method is highly reliable.

F.3 Results on CWQ

In response to the concern that the 1.9% improvement over SIE is close to the standard deviation, we conduct more fine-grained tests on the CWQ dataset:

- SIE: $84.79 \pm 1.03\%$ Hit@1
- TrustFV (Ours): $87.62 \pm 1.7\%$ Hit@1

All three tests uniformly confirm high statistical significance (all $p < 0.001$):

1. Paired t-test: statistic = 6.73, $p < 0.001$
2. Wilcoxon signed-rank test: $p < 0.001$
3. Sample-level win/tie/loss: 96 wins / 23 losses / 3412 ties, binomial test $p < 0.001$

The results demonstrate that the performance improvement is not attributable to random variance, and TrustFV consistently outperforms or matches SIE on the vast majority of individual samples.